České vysoké učení technické v Praze
Fakulta stavební

155UZPD: Semestrální projekt

MMDOS - Síťové analýzy PID

Únor 2019 Petra Mariana Millarová, Michael Kala

1 Cíl projektu

Cílem semestrálního projektu bylo vytvoření konzolové aplikace MMDOS, jež provádí síťové analýzy nad daty PID za využití pgRouting, extenze PostGISu. Uživatel má možnost zadat výchozí a cílovou adresu, aplikace vyhledá nejbližší zastávky hromadné dopravy, provede síťovou analýzu a vrátí nejkratší cestu. Uživateli se poté zobrazí seznam zastávek a linek, které by měl po cestě využít.

2 Data

- Adresní místa RÚIAN Hlavního města Prahy, staženo z Nahlížení do KN [1] (dokumentace: http://vdp.cuzk.cz/vymenny_format/csv/ad-csv-struktura.pdf)
- Síť tras linek Pražské integrované dopravy, staženo z portálu Opendata Hlavního města Prahy [2] (dokumentace: http://www.geoportalpraha.cz/cs/fulltext_geoportal?id=6F576389-385E-4E38-831A-8DE6EFB52C3A#.XErv8stKiV4)
- Zastávky PID - jednotlivé označníky, staženo z portálu Opendata Hlavního města Prahy [2] (dokumentace: http://www.geoportalpraha.cz/cs/fulltext_geoportal?id=63EF19FE-C2FB-4FC2-8C2D-EEBB72C6B81A#.XErxJ8tKiV4)

3 Zpracování dat

3.1 Adresní místa RUIAN

Data byla stažena ve formátu `csv`. Následně pomocí jazyka AWK byl v shellu soubor upraven, aby obsahoval pouze potřebná data (resp. byl vytvořen nový soubor):

```
awk -F \; '{print $11";"$13";"$14";"$15";"$17";"$18}' 20181231_OB_554782_ADR.csv > ruian_adr.csv
```

Ponechány byly sloupce ulice, číslo domovní, číslo orientační, číslo orientační znak (a, b..) a souřadnice x, y v S-JTSK.
Dále byla v databázi na geo102 vytvořena a naplněna tabulka *adr* (včetně geometrie) pomocí dávkového souboru *adr.sql* puštěného pomocí příkazu:

```
psql -h geo102.fsv.cvut.cz -d pgis_uzpd -U uzpd18_a -f adr.sql
```

Obsah dávkového souboru:

```
CREATE TABLE adr(
  gid serial NOT NULL,
  ulice VARCHAR(50),
  c_domovni INTEGER,
  c_orientacni INTEGER,
  co_znak VARCHAR(2),
  y REAL,
  x REAL,
  geom geometry);

\copy adr(ulice, c_domovni, c_orientacni, co_znak, y, x) FROM '../ruian_adr.csv' DELIMITER ';' CSV HEADER encoding 'windows-1250';

UPDATE adr SET geom = ST_GeomFromText('POINT(-'||y||' -'||x||')',5514);

DELETE FROM adr WHERE geom IS NULL;
```

Aby byly názvy ulic importovány správně s diakritikou, bylo třeba nastavit kódování na uvedenou hodnotu `windows-1250`. Zároveň byla pro potřeby projektu z dat vymazána adresní místa bez geometrie.

3.2 Data PID

Data zastávek a tras linek PID byla stažena ve formátu *shp*. Pro import dat do databáze byl využit nástroj PostGISu *shp2pgsql*, jež konvertuje soubory ve formátu *shp* do databázových tabulek. Toto bylo vykonáno pomocí příkazu (v shellu):

```
shp2pgsql -s 5514 -D -I DOP_PID_TRASY_TS_L.shp | psql -h geo102.fsv.cvut.cz -d pgis_uzpd -U uzpd18_a
```

A přímo v databázi byl změněn název vytvořené tabulky:

```
ALTER TABLE dop_pid_trasy_ts_l RENAME TO trasy;
```

Pro potřeby našeho projektu nebyly nutné údaje o nočních linkách, proto byly tyto řádky smazány:

```
DELETE FROM trasy
WHERE (l_metro_n IS NULL
```

```
  OR l_tram_n IS NOT NULL
  OR l_bus_n IS NOT NULL
  OR l_lan_n IS NOT NULL
  OR l_vlak_n IS NOT NULL
  OR l_lod_n IS NOT NULL)
AND l_metro IS NULL
AND l_tram IS NULL
AND l_bus IS NULL
AND l_lan IS NULL
AND l_vlak IS NULL
AND l_lod IS NULL;
```

Analogicky se postupovalo u zastávek:

```
shp2pgsql -s 5514 -D -I DOP_PID_ZASTAVKY_TS_B.shp| psql -h geo102.fsv.cvut.
    cz -d pgis_uzpd -U uzpd18_a
```

```
ALTER TABLE dop_pid_zastavky_ts_b RENAME TO zastavky;
```

U zastávek byly taktéž smazány řádky obsahující pouze údaje o nočních linkách:

```
DELETE FROM zastavky
WHERE zast_denno = 2;
```

Vzhledem k prapodivnosti sloupce `zast_id`, kde se ukázalo, že více zastávek s různými názvy a různým umístěním mají ID stejné, se autoři rozhodli použít sloupec `zast_uzel` jako identifikátor zastávky, protože byla hodnota stejná pro všechny položky se stejným názvem zastávky. Sloupec byl přetypován následujícím příkazem:

```
ALTER TABLE zastavky
ALTER COLUMN zast_uzel_ TYPE INTEGER
USING CAST(zast_uzel_ AS INTEGER);
```

4 Topologie

Vzhledem ke komplikacím zmiňovaným v předchozí kapitole se autoři rozhodli si vytvořit topologii bez použití funkce `pgr_createTopology`. Nejprve byly do tabulky `trasy` přidány sloupce `source` a `target` a vytvořen prostorový index:

```
ALTER TABLE trasy ADD COLUMN "source" integer;
ALTER TABLE trasy ADD COLUMN "target" integer;
CREATE INDEX ON trasy USING gist(geom);
```

Následně byl vytvořen skript v Pythonu, který pomocí prostorového dotazu k začátku i konci každé linie v tabulce `trasy` vybral nejbližší bod z tabulky `zastavky` a vrátil ID jeho uzlu, které pak bylo dosazeno do sloupce `source`, případně `target`.
Prostorový dotaz pro počáteční bod:

```
SELECT DISTINCT ON(t.gid) t.gid, z.zast_uzel_ FROM trasy t, zastavky z
    WHERE ST_DWithin(ST_StartPoint(ST_LineMerge(t.geom)), z.geom, 500) AND t
    .zast_id_od = z.zast_id
```

A pro koncový bod:

```
SELECT DISTINCT ON(t.gid) t.gid, z.zast_uzel_ FROM trasy t, zastavky z
    WHERE ST_DWithin(ST_EndPoint(ST_LineMerge(t.geom)), z.geom, 500) AND t.
    zast_id_ka = z.zast_id")
```

Následně byla vytvořena tabulka vertexů pomocí funkce *pgr_createVerticesTable* a sloupec `the_geom` přejmenován na `geom`:

```
SELECT pgr_createVerticesTable('trasy','geom','source','target');
ALTER TABLE trasy_vertices_pgr rename column the_geom to geom;
```

5 Konzolová aplikace

Pro potřeby konzolové aplikace byly vytvořeny následující SQL funkce:

```
CREATE OR REPLACE FUNCTION FindStationName(id INTEGER)
RETURNS VARCHAR AS $name$
declare
  name varchar;
BEGIN
  SELECT z.zast_nazev INTO name FROM zastavky z
  WHERE z.zast_uzel_ = id LIMIT 1;
  RETURN name;
END;
$name$ LANGUAGE plpgsql;
```

Tato funkce vrací název zastávky podle zadaného ID ze `zast_uzel_`.

```
CREATE OR REPLACE FUNCTION FindVertexID(cd INTEGER, co INTEGER, u VARCHAR)
RETURNS INTEGER AS $id$
declare
  id integer;
BEGIN
  SELECT v.id INTO id FROM adr a, trasy_vertices_pgr v
  WHERE a.c_domovni = cd
  AND a.c_orientacni = co
  AND a.ulice = u
  ORDER BY (a.geom)<->(v.geom) asc limit 1;
  RETURN id;
END;
$id$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION FindVertexIDcd(cd INTEGER, u VARCHAR)
RETURNS INTEGER AS $id$
declare
  id integer;
BEGIN
  SELECT v.id INTO id FROM adr a, trasy_vertices_pgr v
  WHERE a.c_domovni = cd
  AND a.ulice = u
  ORDER BY (a.geom)<->(v.geom) asc limit 1;
  RETURN id;
END;
$id$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION FindVertexIDori(co INTEGER, u VARCHAR)
RETURNS INTEGER AS $id$
declare
  id integer;
BEGIN
  SELECT v.id INTO id FROM adr a, trasy_vertices_pgr v
```

```
  WHERE a.c_orientacni = co
  AND a.ulice = u
  ORDER BY (a.geom)<->(v.geom) asc limit 1;
  RETURN id;
END;
$id$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION FindVertexIDst(u VARCHAR)
RETURNS TABLE(
id BIGINT,
ul VARCHAR,
cp INTEGER,
co INTEGER
  ) as $$
BEGIN
  RETURN QUERY SELECT v.id, a.ulice, a.c_domovni, a.c_orientacni FROM adr a
    , trasy_vertices_pgr v
  WHERE a.ulice = u
  ORDER BY (a.geom)<->(v.geom) asc limit 1;
 END;
$$
LANGUAGE plpgsql;
```

Tyto funkce vrací ID zastávky (`zast_uzel_` prostřednictvím vertexové tabulky), která je nejblíže zadanému adresnímu bodu. První tři funkce jsou uzpůsobeny na zadání kombinace čísla orientačního, domovního, nebo obou a názvu ulice a poslední funkce umožňuje uživateli zadat jen název ulice, přičemž funkce vrací i adresní bod, ke kterému bylo hledání nejbližší zastávky vztaženo.

Následně byl napsán samotný skript aplikace.

6 Závěr

Vytvořili jsme konzolovou aplikaci pro hledání spojů PID, do které lze zadat adresu či její část a aplikace vrací nejkratší trasu včetně linek a přestupů. Námětem na vylepšení je například odlišné ocenění tras metra, tramvají a autobusů, přidání možnosti cestovat v noci, přidání jízdních řádu atp.
Aplikace samozřejmě nemůže nahradit veřejné služby jako je iDoS nebo vyhledávač spojení od Dopravního podniku hlavního města Prahy, do kterých bylo vloženo o mnoho více času a úsilí.

7 Zdroje

[1] Nahlížení do KN, aplikace ČÚZK - `https://nahlizenidokn.cuzk.cz/`
[2] Portál pro Otevřená data hlavního města Prahy - `http://opendata.praha.eu/`
[3] Tvoření topologie pomocí PGRouting - `http://docs.pgrouting.org/latest/en/pgr_createTopol`
[4] Tvorba nové tabulky vertexů - `http://docs.pgrouting.org/latest/en/pgr_createVerticesTabl`
[5] Funkce pro vyhledání nejkratší trasy - `http://docs.pgrouting.org/latest/en/pgr_dijkstra.htm`